**tפרק 2 : המודל הראשון שלי**

פרק זה ירגיש לחלקכם כמו hello world ל-machine learning, וכך הוא צריך להרגיש. ניקח בתור התחלה מודל פשוט (ומעצם כך ניתן להגיד עליו דברים יפים אנליטית שלא ניתן להגיד על מודלים יותר מורכבים בכזו אלגנטיות) וננסה להשתמש בו על בעיה אמיתית.

ההתחככות הזו של להפעיל מודל במציאות מעוררת הרבה שאלות, כיצד נכון לבחון את המודל, איך משווים מודלים שונים, ואילו בורות מחכים לי - איש הדאטא סיינס המתחיל.

**פרק זה יתעסק בבעיות רגרסיה - שהן אולי הצד היותר אינטואיטיבי להגדרת loss והבנת הframework הסטנדרטי.**

**חלק תיאורטי:**

1. קרא את פרק 3 בISLR - רגרסיה לינארית - Linear Regression.
2. קרא את פרק 5 בISLR - שיטות חלוקת דאטא - Resampling Methods.
3. קרא את פרקים 6.1 ו6.2 בISLR - השוואת מודלים - Linear Model Selection.

**חלק מעשי:**

בקהילה של הדאטא סיינס יש מספר דאטאסטים מוכרים אשר מהווים בנצ'מארק לכלל הקהילה. מאגר המידע על מחירי שכונות בבוסטון הוא אחד מהם.

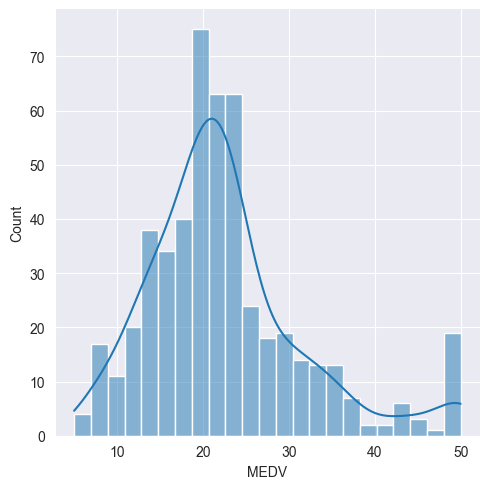
המאגר מכיל לא הרבה רשומות, על שכונות בבוסטון ומחירי הדיור הממוצע בהם. על כל שכונה ישנם מספר פריטי מידע אשר יכולים לעזור לחזות את המחיר הממוצע בשכונה.

המאגר הוא בין הדאטאסטים הראשונים שנוצרו בתחום, הוא אינו גדול ומשום כך שאלת התאמת היתר עולה בצורה יותר חדה.

היעזר בsklearn על מנת לטעון את המידע למחברת.

**לאחר מכן בצע את שלבי המחקר הקלאסי (כמובן שזו רק שבלונה, ויש לעשות איטרציות תכופות עם החונך בשלב זה, ולחזור על שלבים קודמיםלאחר הערות):**

1. בצע אקספלורציה: לדוגמא -
   1. בחן התפלגויות וקורלציות



We can clearly see clear skewness to the right.

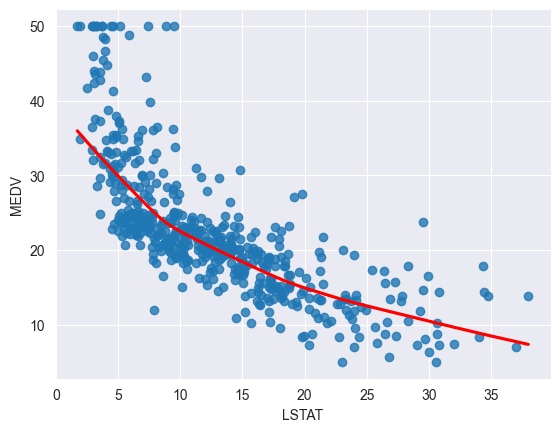
Lets look at the different kinds of correlations to check collinearity:

Add correlations plot (for example do absolute value for correlations and then plot in a bar plot)

pearson\_correlation = boston\_df.corr('pearson')["MEDV"].sort\_values(ascending=False)  
kendall\_correlation = boston\_df.corr('kendall')["MEDV"].sort\_values(ascending=False)  
spearman\_correlation = boston\_df.corr('spearman')["MEDV"].sort\_values(ascending=False)

After checking the correlations, we see LSTAT has a pretty strong negative correlation across all kinds of correlations. LSTAT is the % of lower status of the population ( which I think means poor people). This makes sense because the higher the median of a owner-occupied home, the harder buying new homes becomes for poor people.

Lets visualize them together:



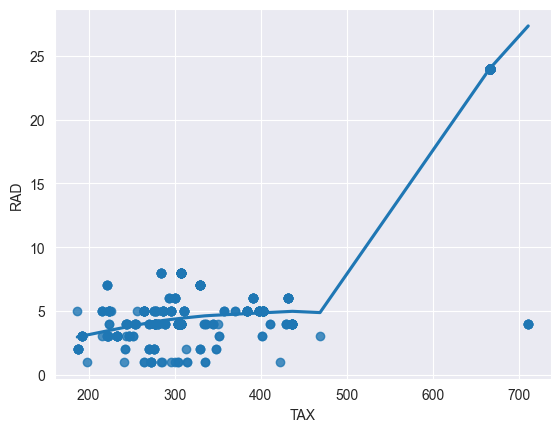
Its clear the data is monotonic, although not linear, so to check if a relationship actually exist, we will calculate a Spearman correlation coefficient with associated p-value. We get the following results:

Corr = -0.8529141394922163

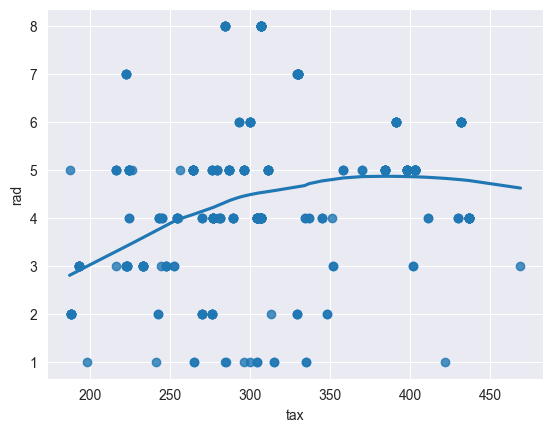
p-value = 2.221727524313283e-144

This means we can reject the null hypothesis that means there is no relationship between the parameters.

Lets look at another case, where we can see where high correlation can be wrong:

For the features TAX and RAD, there is a pearson correlation of 0.91 and a very small p-value, lets visualize:  


Its clear that they DO NOT have a linear relationship, even though we have evidence that suggests otherwise. This happens because of the outliers. Lets remove them and check again:



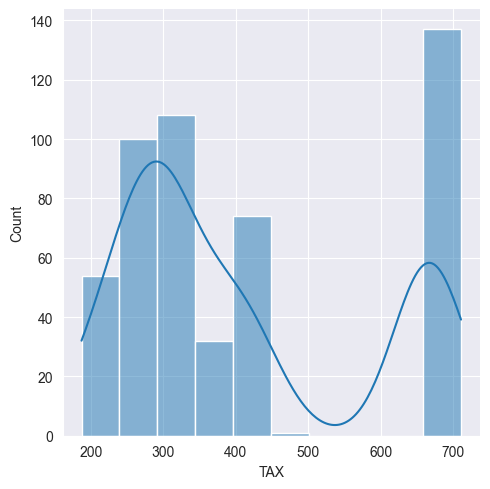
Now we get more reasonable results:

Corr = 0.21

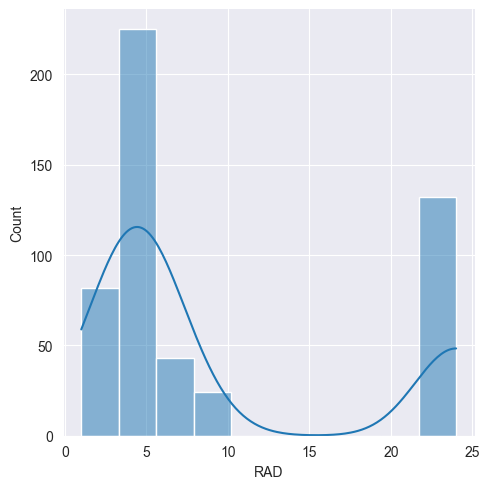
p-value = 0.0000363 (still a small p-value)

BUT, removing the outliers is wrong! We need them, I had to remove 132 samples that were supposedly “outliers” because they all looked very far away.

If we look at the distplot we can see some better idea about how TAX really looks:



We look also at the distribution of RAD:



Now we understand that the “outliers” are VERY important to see the entire image.

This suggests we might want to split this feature.

* 1. וודא שאתה מבין את משמעות העמודות - מתיאורן ומהערכים שהם מכילים

This dataset is used for predicting the median housing price in different areas in boston. It contains different features like crime rate, is the area next to the river, distance to employment areas and more.

Using:

print(boston.DESCR)

We can get some information about the variables:

CRIM per capita crime rate by town

ZN proportion of residential land zoned for lots over 25,000 sq.ft.

INDUS proportion of non-retail business acres per town

CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

NOX nitric oxides concentration (parts per 10 million)

RM average number of rooms per dwelling

AGE proportion of owner-occupied units built prior to 1940

DIS weighted distances to five Boston employment centres

RAD index of accessibility to radial highways

TAX full-value property-tax rate per $10,000

PTRATIO pupil-teacher ratio by town

B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town

LSTAT % lower status of the population

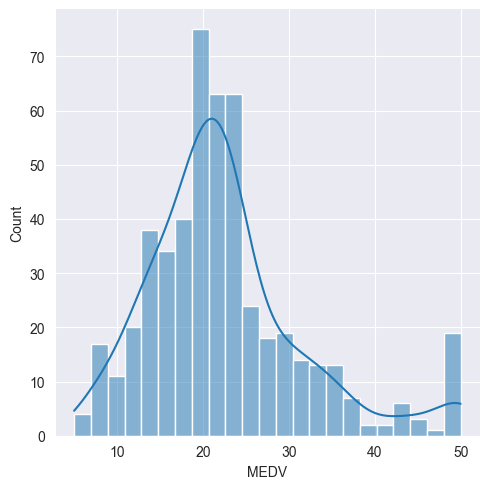
MEDV Median value of owner-occupied homes in $1000's

All features are quantitative except RAD and CHAS

.

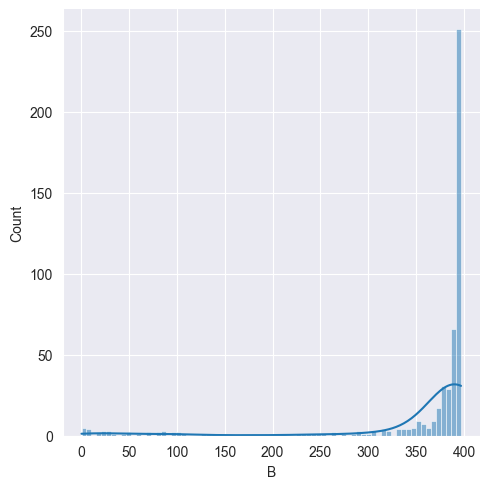
**Anomalies and Outliers:**

Firstly, lets check the target feature. The target feature has a bell shaped distribution, so we can find anomalies by checking which values are 3 stds away from the mean (less then 99.7% chance of existing). After checking, I see there aren’t any values like that.



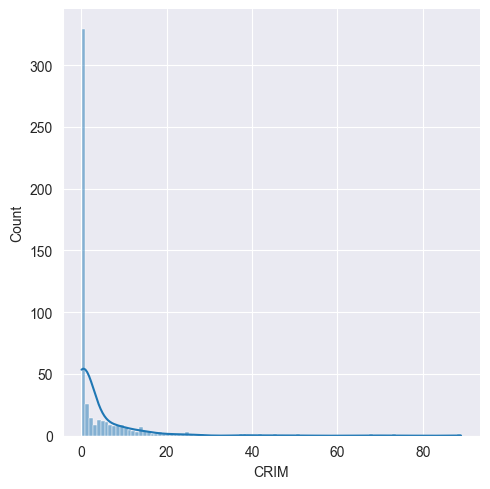
Then, we look at all the different distributions, we can clearly see

some outliers in the B, CRIM and ZN features. Lets start by looking at B:



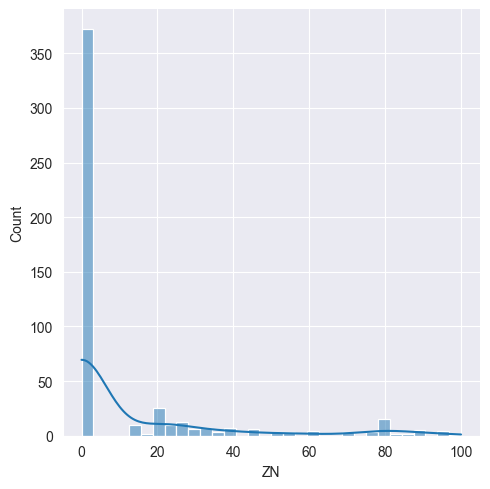
We can see almost the entire data is located above 300. When we think about it, its very possible some neighborhoods have less black people around them, simply because in the united states black people are a minority.

Lets look at CRIM:



If we looks at the crime rates(CRIM) above, we can see most neighborhoods have a crime rate of less then 25, with one outlier that reach up to 88% crime rate! Even though this might happen, this outliers can add significant bias to our model.

Lets look at ZN:



This shows the percentage of residents with land bigger then 25,000 sq.ft, its very clear the distribution is heavily skewed to the right. Most neighborhoods have no resident with big properties, but some neighborhoods have some rich people that do have a big property. This suggests we might want to split this feature to handle cases where the value is 0 vs values that aren’t 0.

**Training**

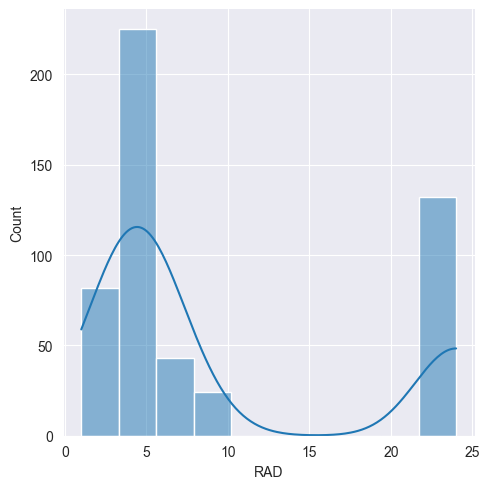
When using a non-processed model which we will use as baseline, the results are:

R² Score: 0.5892223849182527

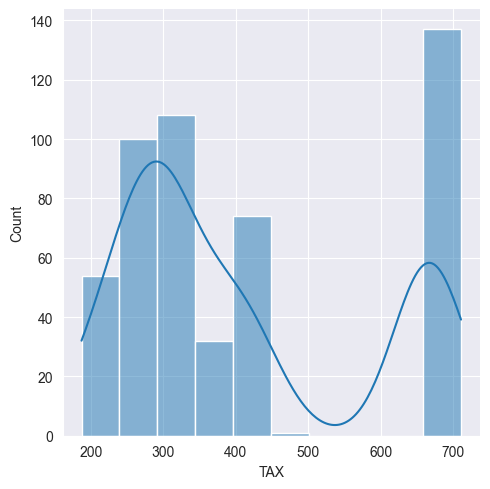
MSE: 33.448979997676375

Lets try using the suggestions we had previously, where we split some features that are heavily skewed:

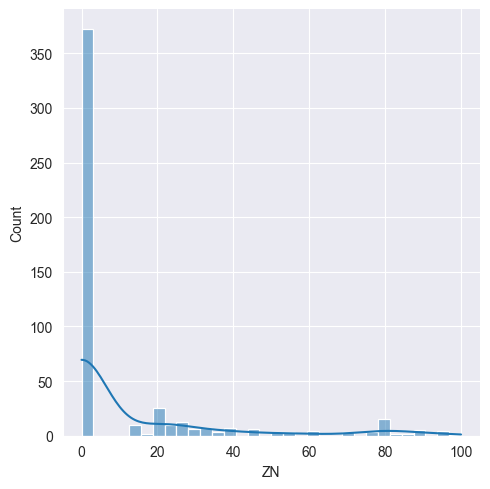
* + - 1. We split RAD at the average of the two biggest values:



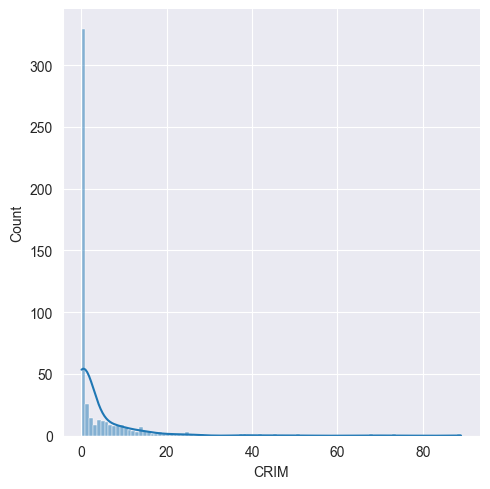
* + - 1. We split TAX at about 600, which makes sense according to the distribution:



* + - 1. We split ZN at 1:

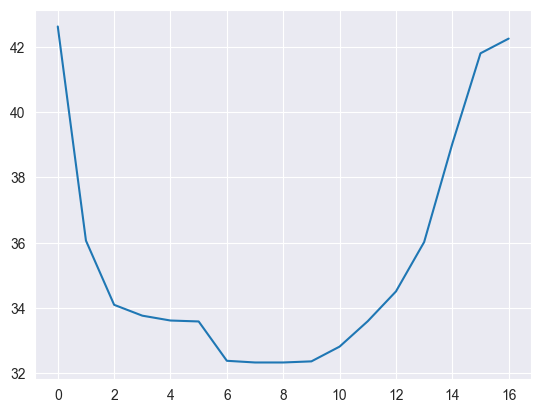


* + - 1. Finally, we split CRIM to NO crim at all and some CRIM:

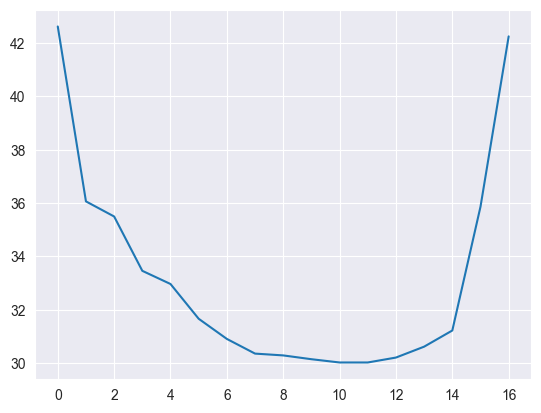


Now, we have 16 features instead of 13, so we need to pick only the relevant ones. We cant use an exhaustive feature selector because we have too many features, so we use a sequential feature selector. We use both a forward and a backward method:

Forward:



Backward:



We see that around 10-11 features is about the sweet spot for the MSE, but lets try improving it even more.

After that, we do MinMax scaling so we stay as close as we can to the original distribution, while also achieving numerical stability.

We then do some feature engineering, and add some polynomial features. We do this while simultaneously trying both ridge linear regression and lasso linear regression. For a degree of 2 polynomials we get the following results:

Test MSE (Ridge, Polynomial): 27.319110958754546

Test R2 (Ridge, Polynomial): 0.6274689538513338

Test MSE (Lasso, Polynomial): 27.303443716380375

Test R2 (Ridge, Polynomial): 0.6276825967550406

While if we go further up to a degree of 5 polynomials:

Test MSE (Ridge, Polynomial): 24.571307589665434

Test R2 (Ridge, Polynomial): 0.6649387699534411

Test MSE (Lasso, Polynomial): 24.84643548543118

Test R2 (Lasso, Polynomial): 0.6649387699534411

**In conclusion, we improved our model by 27%!**